REST API

Building Modern Web Services with PHP

What is REST API?

REST = **RE**presentational **S**tate **T**ransfer

REST lets programs talk to each other over the internet using standard web methods: **GET, POST, PUT, DELETE**.

Real-world Analogies

- Library
- Restaurant

Library Analogy

- The internet = a big **library**
- Each item (profile, photo, post) = a **resource** with an address (URL)
- REST methods:
  - **GET** → read data (ask for a book)
  - **POST** → create new (add a book)
  - **PUT** → update existing (fix a book)
  - **DELETE** → remove (take out a book)

- The server replies with **data** or a **status** (e.g., `"OK"`, `"Not Found"`).
- RESTful APIs use simple rules so systems can easily work together.
- REST is like a **common language** for apps to communicate clearly over the web.

Restaurant Analogy

REST API is like a **restaurant menu and ordering system**:

- **Menu** = Available endpoints (what you can order)
- **Ordering** = Making requests (GET, POST, PUT, DELETE)
- **Kitchen** = Server processing your request
- **Food delivery** = Response with data

Why REST API is Important

1. Separation of Frontend and Backend

```
Mobile App    →  \
Web App       →   → REST API → Database
Desktop App   →  /
IoT Device    →
```

One API serves multiple applications!

2. Scalability

- Frontend and backend teams work independently
- Easy to update UI without changing backend
- Can handle millions of users

3. Industry Standard

- Used by **every major company**
- Essential skill for web developers
- Enables modern app architecture

4. Reusability

```
// Same API endpoint used by:
GET /api/users/123

// Web app: Display user profile
// Mobile app: Show user info
// Admin panel: User management
// Third-party apps: Integration
```

5. Technology Independence

- **Frontend**: React, Vue, Angular, Flutter, iOS, Android
- **Backend**: PHP, Node.js, Python, Java, .NET
- **Database**: MySQL, PostgreSQL, MongoDB

They all communicate via REST API!

REST API Principles

1. Stateless

Each request contains all the information needed.

```
GET /api/users/123
Authorization: Bearer token123
```

2. Resource-Based URLs

```
/api/users          → All users
/api/users/123      → Specific user
/api/users/123/posts → User's posts
```

3. HTTP Methods

- **GET**: Retrieve data
- **POST**: Create new data
- **PUT**: Update existing data
- **DELETE**: Remove data

Real-World REST API Examples (API Servers)

Companies Using REST APIs

- **Google Maps**: Location
- **Twitter**: Tweet posting and reading
- **GitHub**: Repository Management
- **Facebook**: Social media integration
- **Stripe**: Payment
- **YouTube**: Video Streaming

And much more, including:

- **Amazon**: E-commerce and AWS
- **Uber**: Ride booking and tracking
- **Spotify**: Music streaming

1. Google Maps API

```
// Get location information
GET https://maps.googleapis.com/maps/api/geocode/json?address=Seoul

// Response:
{
  "results": [{
    "formatted_address": "Seoul, South Korea",
    "geometry": {
      "location": {"lat": 37.566535, "lng": 126.9779692}
    }
  }]
}
```

Used by: Uber, food delivery apps, real estate websites

2. Twitter API

```
// Get user's tweets
GET https://api.twitter.com/2/users/by/username/elonmusk

// Post a tweet
POST https://api.twitter.com/2/tweets
{
  "text": "Hello Twitter API!"
}
```

- We can tweet without the Twitter app.

3. GitHub API

```
// Get repository information
GET https://api.github.com/repos/microsoft/vscode

// Create a new repository
POST https://api.github.com/user/repos
{
  "name": "my-new-project",
  "description": "My awesome project"
}
```

4. Stripe Payment API

```
// Process payment
POST https://api.stripe.com/v1/charges
{
  "amount": 2000,
  "currency": "usd",
  "source": "tok_visa"
}
```

5. YouTube Data API

```
// Search videos
GET https://www.googleapis.com/youtube/v3/search?q=programming

// Get video details
GET https://www.googleapis.com/youtube/v3/videos?id=VIDEO_ID
```

These APIs power thousands of applications!

Career Relevance

- **Essential skill** for modern web development
- **High demand** in job market
- **Foundation** for microservices architecture
- **Enables full-stack development**

You now understand the backbone of modern web applications!